



SSE | Tutorial 2

Task 1

Get informed about the Proxy software design pattern¹. Create a proxy for the Calculator class from the 1st tutorial. The proxy should cache last 10 calculations and return the cached results when the new and the cached operation/operands pairs match. The Task1-Template.zip gives you a good starting point if you have not finished the 1st tutorial.

Task 2

Answer the following questions:

- What is Middleware?
- Which services are usually provided by Middleware?
- What is Remote Procedure Call?

Task 3

- What is Marshalling?
- The *Marshalling.zip* project implements a scenario, in which a client sends an object to a server, which formats the object and prints to console. The connection is established by middleware, which in addition takes care of transfer of typed objects.
 - Read the code and learn how the connection gets established.
 - Implement the methods *Marshal* and *Demarshal*
 - Start the two projects *Client* and *Server* and test your implementation (right click on *Solution* in *Solution Explorer* and then *StartUp Projects*)
- Beside the default printing functionality, the server provides two string operations:
 - string** concat(**string** arg1, **string** arg2, **string** arg3)
 - string** substring(**string** str, **string** positionIndex)

Assume, there are more of such operations, all only with string parameters. Extend your client towards RPC of these operations. Complete the placeholders marked with TODO. Define a message encoding scheme for RPC calls and use Reflection² on the server side to invoke the methods.

¹ For example, here: <http://www.oodeign.com/proxy-pattern.html>

² <http://www.codeproject.com/Articles/17269/Reflection-in-C-Tutorial>

Task 4: Homework

- a) Describe the architecture of the Component Object Model (COM). What are the tasks of COM Component, COM Server, COM Registry and COM Interface?
- b) Extract the project *COM.zip*. The subproject *ComBrowserServer* implements a COM component, which makes HTTP-GET requests to a given URL. *ComBrowserClient* is a C++ application, which makes use of the COM component.
 - Compile both projects
 - Register the component *ComBrowser.dll* in the registry of your OS (as described in *ComBrowserServer.cs*)
 - Start the subproject *ComBrowserClient* and test if the component is called